Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended): A method for handling resets at a bus bridge, the method comprising:

providing ~~receiving a reset signal at~~ a bus bridge coupling a master component supporting a master component bus protocol to a slave component supporting a slave component bus protocol, wherein either the master component or the slave component is implemented in programmable logic;

receiving a reset signal at the bus bridge, the reset signal associated with a reset of the slave component; and

responding to a transaction from the master component to the slave component with a bus protocol compliant signal after the reset signal is received, wherein responding with the ~~a~~ bus protocol compliant signal allows the master component to continue operating while the slave component is being reset.

2. (Original): The method of claim 1, wherein the reset signal is received at a bus bridge master interface.

3. (Original): The method of claim 1, wherein a bus bridge slave interface responds with the bus protocol compliant signal.

4. (Currently Amended): A ~~The~~ method ~~of claim 1~~ for handling resets at a bus bridge, the method comprising:

receiving a reset signal at a bus bridge coupling a master component supporting a master component bus protocol to a slave component supporting a slave component bus protocol, the reset signal associated with a reset of the slave component; and

responding to a transaction from the master component to the slave component with a bus protocol compliant signal after the reset signal is received, wherein responding with the bus protocol compliant signal allows the master component to continue operating while the slave component is being reset, wherein the bus protocol compliant signal is an AHB bus error signal.

5. (Original): The method of claim 1, wherein the master component bus protocol is the AHB protocol operating in a master component clock domain.

6. (Original): The method of claim 5, wherein the slave component bus protocol is the AHB protocol operating in a slave component clock domain different from the master component clock domain.

7. (Original): The method of claim 5, wherein the slave component bus protocol is the PI protocol.

8. (Original): The method of claim 5, further comprising synchronizing the reset signal to the master component clock domain.

9. (Original): The method of claim 8, further comprising setting the master component to an idle state.

10. (Currently Amended): The method of claim 9, further comprising setting ~~the~~ a bus bridge slave interface in an idle state while the slave component is being reset.

11. (Currently Amended): A method for independently resetting a side of a bus bridge, the method comprising:

receiving a reset signal on a first side of the bus bridge such that a component associated with the first side is reset while a component associated with a second side of the bus bridge continues operating without violating bus protocols, the first side corresponding to a first clock domain and the second side corresponding to a second clock domain, wherein either the component associated with the first side or the component associated with the second side is implemented in programmable logic.

12. (Original): The method as recited in claim 11, further comprising:

synchronizing the reset signal with the second side of the bus bridge such that a component associated with the second side can transition to an idle state.

13. (Currently Amended): The method as recited in claim 11, wherein receiving the reset signal includes:

receiving the reset signal at an ~~first~~ interface, the ~~first~~ interface coupling the first side component to the bus bridge.

14. (Currently Amended): A ~~The~~ method ~~as recited in claim 13~~ for independently resetting a side of a bus bridge, the method comprising:

receiving a reset signal on a first side of the bus bridge such that a component associated with the first side is reset while a component associated with a second side of the bus bridge continues operating without violating bus protocols, the first side corresponding to a first clock domain and the second side corresponding to a second clock domain,

wherein receiving the reset signal ~~further~~ includes~~:~~ synchronously receiving the reset signal at both an ~~the first~~ interface and an alternating state handler, the interface coupling the first side component to the bus bridge and the alternating state handler being associated with the first side.

15. (Original): The method as recited in claim 14, wherein the alternating state handler is an SR type flip-flop.

16. (Currently Amended): The method as recited in claim 11, wherein both the components are ~~associated with the first side is selected from the group consisting of a programmable logic device, a hard logic device, and a combination of both~~ implemented in programmable logic.

17. (Currently Amended): The method as recited in claim 12, wherein one of the components ~~associated with the second side~~ is ~~selected from the group consisting of a processor core, a hard logic device, and a combination of both~~ implemented in hard logic.

18. (Currently Amended): The method as recited in claim 12~~3~~, wherein synchronizing the reset signal includes:

receiving a synchronization signal by synchronization logic, the synchronization signal being a result of the reset signal;

receiving a clock input by the synchronization logic, the clock input associated with the second clock domain; and

outputting an initialization idle signal when the received clock input indicates that the second clock domain is running.

19. (Currently Amended): A ~~The~~ method ~~as recited in claim 18, further comprising~~ for independently resetting a side of a bus bridge, the method comprising:

receiving a reset signal on a first side of the bus bridge such that a component associated with the first side is reset while a component associated with a second side of the bus bridge continues operating without violating bus protocols, the first side corresponding to a first clock domain and the second side corresponding to a second clock domain; and

synchronizing the reset signal with the second side of the bus bridge such that a component associated with the second side can transition to an idle state, wherein synchronizing the reset signal includes:

receiving a synchronization signal by synchronization logic, the synchronization signal being a result of the reset signal;

receiving a clock input by the synchronization logic, the clock input associated with the second clock domain; and

outputting an initialization idle signal when the received clock input indicates that the second clock domain is running;

receiving the initialization idle signal by an ~~second~~ interface~~, the second interface~~ configured for coupling to the component associated with the second side of the bridge; and

outputting a component idle signal to the component associated with the second side of the bridge in response to receiving the initialization idle signal, the component idle signal being used to trigger the component associated with the second side of the bridge to transition to the ~~an~~ idle state.

20. (Currently Amended): The method as recited in claim 19, further comprising:

outputting a confirmation idle signal from the ~~second~~ interface, the confirmation idle signal provided to indicate when the component associated with the second side of the bridge has reached the idle state.

21. (Currently Amended): A bus bridge for handling resets, the bus bridge comprising:

a bus bridge master interface, the bus bridge master interface coupled to a slave component, the bus bridge master interface configured to receive a reset signal associated with a reset of the slave component on a first side of the bus bridge;

a bus bridge slave interface, the bus bridge slave interface coupled to a master component, the bus bridge slave interface configured to interact with the master component on a second side of the bus bridge such that the slave component is reset while the master component continues operating without violating bus protocols, wherein either the master component or the slave component is implemented in programmable logic.

22. (Original): The bus bridge as recited in claim 21, wherein the first side corresponds to a first clock domain and the second side corresponds to a second clock domain, the first and second clock domains being different from each other.

23. (Original): The bus bridge as recited in claim 22, further comprising:

a synchronization logic for synchronizing the reset signal with the second clock domain such that the master component and the bus bridge slave interface transition to an idle state.

24. (Currently Amended): A ~~The~~ bus bridge ~~as recited in claim 21, further comprising~~ for handling resets, the bus bridge comprising:

a bus bridge master interface, the bus bridge master interface coupled to a slave component, the bus bridge master interface configured to receive a reset signal associated with a reset of the slave component on a first side of the bus bridge;

a bus bridge slave interface, the bus bridge slave interface coupled to a master component, the bus bridge slave interface configured to interact with the master component on a second side of the bus bridge such that the slave component is reset while the master component continues operating without violating bus protocols; and an alternating state handler being associated with the first side, wherein both the bus bridge master interface and the alternating state handler are configured for synchronously receiving the reset signal on the first side of the bus bridge.

25. (Original): The bus bridge as recited in claim 24, wherein the alternating state handler is a SR type flip-flop.

26. (Currently Amended): The bus bridge as recited in claim 21, wherein both the ~~slave~~ components are ~~is selected from the group consisting of a programmable logic device, a hard logic device, and a combination of both~~ implemented in programmable logic.

27. (Currently Amended): The bus bridge as recited in claim 21, wherein one of the ~~master~~ components is ~~selected from the group consisting of a processor core, a hard logic device, and a combination of both~~ implemented in hard logic.

28. (Original): The bus bridge as recited in claim 23, wherein the synchronization logic is configured for receiving a synchronization signal that is a result of the reset signal; for receiving a clock input that is associated with the second clock domain; and for outputting an initialization idle signal when the received clock input indicates that the second clock domain is running.

29. (Currently Amended): A ~~The~~ bus bridge ~~as recited in claim 28~~ for handling resets, the bus bridge comprising:

a bus bridge master interface, the bus bridge master interface coupled to a slave component, the bus bridge master interface configured to receive a reset signal associated with a reset of the slave component on a first side of the bus bridge;

a bus bridge slave interface, the bus bridge slave interface coupled to a master component, the bus bridge slave interface configured to interact with the master component on a second side of the bus bridge such that the slave component is reset while the master component continues operating without violating bus protocols, wherein the first side corresponds to a first clock domain and the second side corresponds to a second clock domain, the first and second clock domains being different from each other;

a synchronization logic for synchronizing the reset signal with the second clock domain such that the master component and the bus bridge slave interface transition to an idle state, wherein the synchronization logic is configured for receiving a synchronization signal that is a result of the reset signal; for receiving a clock input that is associated with the second clock domain; and for outputting an initialization idle signal when the received clock input indicates that the second clock domain is running, wherein the bus bridge slave interface is configured for receiving the initialization idle signal; and for outputting a component idle signal to the master component in response to receiving the initialization idle signal, the component idle signal being used to trigger the master component to transition to an idle state.

30. (Original): The bus bridge as recited in claim 29, wherein the bus bridge slave interface is further configured for outputting a confirmation idle signal that is used to indicate when the master component has reached an idle state.

31. (Currently Amended): A computer ~~program product comprising a machine~~ readable recording medium storing ~~on which is provided~~ program instructions for handling resets at a bus bridge, the program instructions comprising:

instructions for receiving a reset signal at a bus bridge coupling a master component supporting a master component bus protocol to a slave component supporting a slave component bus protocol, the reset signal associated with a reset of the slave component; and

instructions for responding to a transaction from the master component to the slave component with a bus protocol compliant signal after the reset signal is received, wherein responding with a bus protocol compliant signal allows the master component to continue operating while the slave component is being reset, wherein either the master component or slave component is implemented in programmable logic.

32. (Currently Amended): The computer readable recording medium ~~program product~~ of claim 31, wherein the reset signal is received at a bus bridge master interface.